Data Analysis Using Stata

Jan 9, 2020

1. Review briefly - Introduction to Stata

Understand and/or know how to:

- Stata windows and main files
- Interact with Stata
- Load non-Stata format data into Stata (mainly Excel files and csv files)
- Variable types, data types, variable naming conventions; missing values
- Convert between numeric variables and string variables
- Generate a categorical/an indicator variable from a continuous/string variable
- Label a variable; define a value label and attach it to a variable

2. Steps in the data analysis:

- Planning the organization of your files and directories
  - systematically think ahead - simple and consistent principles
    - Directories on your hard drive – where to store your course and/or research materials
    - One directory only for each project – so it's easy to find your project files
    - Names for projects, folders and files – all mnemonic
  - **Separate do-files for data management and statistical analysis**
    - Individual do-files: typically (but not always) start with the letters cr or an
    - Or create your own rules for your do-files' names
- Data management: well documented do-files
  - Verify your data are accurate and no duplicate observations; missing values are properly handled
  - Generate the variables that are required for the next step of statistical analysis; make sure all variables are well named and properly labeled
  - …
- **Running statistical analysis**
- Presenting results – interpret your analysis results
- Protecting files – documentation to keep track of what you have done and thought

3. Read/understand help files

- . help language
  - [prefix :] command [varlist] [=exp] [if] [in] [weight] [using filename] [, options]
    - Square brackets are optional; fill in only if needed
    - For example: **. use daisy2red.dta, clear**
      - . **sum**marize
      - . by herd: gen totaltwin = sum(twin) if herd >= 1 & herd < 3
    - . help varlist //or click the link of varlist
      - . help newvarlist
      - . help fvvarlist
      - . help varname
        - ✓ When using the term varname, it usually mean an existing varname -- a variable that already exists in the dataset. So if referring to it, we can abbreviate -- use only some of the leading characters -- as long as we specify enough to uniquely identify the variable:
        - ✓ Myv might be a unique abbreviation for Myvar
- . help regress
  - regress depvar [indepvars] [if] [in] [weight] [, options]
    - . regress milk120

- . regress milk120 parity
- . regress milk120 parity if herd ==1
- . regress milk120 parity in 1/272
- [fweight=pop]
  - ✓ . help weight
- . regress milk120 parity, nocons level(90)

- . help invttail() //. h ln()
  - . display invttail(1534,0.025)
  - . display ln(10) // =. display log(10)

4. Dissect a sample do-file: 1al.do (statistical analysis)

- Robust
- Legible

4.1 A few simple commands: usually at the beginning of a do-file - Robust

- Version control
  - . help version
  - . version 16.0
- Prevent –more- condition
  - . help more
  - . set more off, permanently
- Change working directory – always use it
  - . help cd
  - . cd "r:\"
- Log the output of do-files*
  - . help log
  - . log using assignment1.smcl
- Include seeds for random numbers*
  - . help set seed
  - . set seed 20200108

4.2 Comments, blank lines and long lines in do-files - Legible

- Use lots of comments:
  - . help comments
- Use blank lines
  - Freely include blank lines in your do-file to make your do-files more clear
- Split a long line into several lines to make long lines more readable (. help comments)
  - Use three forward slashes (///) where you want to split the line
  - Use /*
    */ in the two different lines

5. Debugging do-files:

- determining the source of an error: read and try to understand error messages
- Common errors – fix them (may not need to explore the error code)
  - no; dataset in memory has changed since last saved
    - ask yourself whether you want to overwrite the current dataset in memory
  - log file already exists
    - ask yourself whether you want to overwrite the log file
  - incorrect command name:
    - unrecognized command: regriss

- incorrect variable name:
  - variable milkl20 not found
- incorrect option of the command:
  - option clea not found
  - option mksymbol() not allowed
  - option stand not allowed
- missing comma before option:
  - invalid 'clear'
  - too many variables specified
  - variable nocons not found
- not sorted; too many values…

- Tolerate errors - allowing the command/do-file to continue despite errors
  - . capture log close
  - . capture drop rawres stdres delres
    (note: Stata commands do either exactly what you say or nothing at all. If one of the three variables did not exist in the data, drop would then do nothing. It would not drop the other two. To achieve the desired result, we must give three commands:
    . capture drop rawres
    . capture drop stdres
    . capture drop delres
- Other methods to resolve errors
  - Drop all derived/generated variables, trying an alternative approach
  - Run the do-file in steps
  - Reload the data set
  - Restart Stata

6. Create graphs

- Use menus and dialogue boxes
  - . sysuse auto, clear
    . twoway (scatter mpg weight, mcolor(red) msize(medsmall) msymbol(triangle)), ///
    ylabel(, angle(horizontal)) scheme(s1mono) name(mpgwt, replace)
- Use Graph Editor and Graph Play
  - . twoway (scatter mpg weight)
  - Create a Graph Recording file (. help graph editor):
    - Click **Start Graph Editor** on your Stata graph;
    - Click **Start recording,** then start editing your graph….once finish editing your graph;
    - Click **End recording;**
    - Click **browse** and navigate to your project folder, **type a name** (myScatter) to give your Recording File a name, then click Save button;
    - Click **Stop Graph Editor.**
  - To play the recording by point and click:
    - Click **Start Graph Editor** on your Stata graph
    - Click on the **Play Recording button**. You will be presented with a list of your recordings.
    - Select the recording name you want to play, the edits will be applied to your graph;
    - Click **Stop Graph Editor**
  - To play the recording in your do-file:
  - . twoway (scatter mpg weight), play(myScatter) //assume it's in your current working folder

7. A sample do-file for statistical modeling: 1al.do

- Understand the data

  - Understand the data structure
  - Assure the outcome variable and all independent variable(s) are numeric variables
- Describe the variables
  - Understand the distribution of each of the variables on its own
  - Understand the nature and strength of relationships among variables
  - Create graphs
  - Create summaries or distribution tables
- Statistical model
  - regress milk120 parity
- Predictions from the model
  - Acquire model predicted values
  - Generate variables/values required for model forecasting and/or graphs
  - Display/List values
  - Create graph(s)
- Evaluate the model validity: compute model residuals
  - Individual residual checking
  - Overall residual checking
    - Assessing normality and homoscedasticity of residuals
    - Other assessment: residuals versus fitted value - linearity
  - Is the model valid?
    - Yes – interpretation and reporting
    - No – go back to modify the previous model

8. Do-files (again): Robust & Legible

8.1 Robust means:

- A do-file produces exactly the same result when rerunning the do-file at a later time or on different computers (only need to change the working directory)
- A do-file is self-contained:
  - Should not rely on something left in memory by a prior do-file or commands run from the Command Window
  - Should not use a dataset unless it loads the dataset itself
  - Should not compute a test of coefficient unless it estimates those coefficients in the same do-file
- Explanation for the commands:
  - Use version control - Stata might change the command names or computational methods
  - Change directory – exclude directory information or keep minimum directory location in commands that read or write files
  - Include seeds if random numbers required
    - If try to replicate results that use random numbers, you need to use the same random numbers; otherwise you will obtain different results

8.2 Legible means:

- Internally documented and carefully formatted
  - Make the content/logic clear
  - Help you and your supervisor(s)/collaborator(s) easier to debug and understand what you did
- Several practical steps:
  - Use lots of comments
  - Use blank lines
  - Use alignment and indentation if you prefer*
  - Split a long line into several lines (more helpful for Stata 14 and earlier version)